

Detección de Fake News A Través del Algoritmo Naive Bayes

November 7, 2020

Detección de Fake News A Través del Algoritmo Naive Bayes

0.1 Introducción

Las noticias falsas no son un fenómeno nuevo, sin embargo, en la actualidad es más fácil ser víctima de engaño e incluso manipulación, procedente de la velocidad a la que se accede a la información y la facilidad con la que contamos para acceder a decenas de noticias en solo unas horas. Es una problemática que va en ascenso correspondientemente a como el tiempo y la tecnología avanza. Es por eso que en este artículo trataremos de clasificar noticias falsas y reales utilizando un algoritmo de minería de datos denominado Naive Bayes.

0.2 Keywords

Algoritmo, Naive Bayes, Fake News.

0.3 1. Planteamiento del problema

0.3.1 1.1 Descripción del problema

La mayoría de noticias falsas provienen de Estados Unidos que más tarde son traducidas al español. Muchas de ellas se infiltran a nosotros a través de redes sociales. Por ejemplo: El 4 de junio, el sitio de noticias falsas Science Post publicó un bloque de texto “lorem ipsum” bajo un título llamativo: Study: 70% of Facebook users only read the headline of science stories before commenting.

Casi 46.000 personas compartieron la publicación, algunas de ellas con mucha seriedad. Ahora, como si necesitara más pruebas, el titular ha sido validado una vez más: según un nuevo estudio realizado por científicos informáticos de la Universidad de Columbia y el Instituto Nacional Francés, el 59 por ciento de los enlaces compartidos en las redes sociales nunca se ha hecho clic en realidad: en otros En palabras, la mayoría de las personas parecen retuitear noticias sin siquiera leerlas.

0.3.2 1.2 Formulación del problema

Debido a todos los problemas que pueden derivarse de compartir noticias falsas y teniendo en cuenta el hecho de que la mayoría de personas comparte noticias sin, siquiera leer la noticia completa, hemos decidido escribir un programa en Python que es capaz de predecir si una noticia será falsa o verdadera con tan solo el titular de la noticia.

0.3.3 1.3 Variables del problema

SUPOSICIONES ya que, si no hay características que no alcance a identificar o no reconozca, la predicción será errónea.

0.4 2. Objetivos

0.4.1 2.1 Objetivos generales

El objetivo general de la aplicación será entrenar a el algoritmo para poder ser capaz de distinguir entre titulares de noticias falsas y verdaderas.

0.4.2 2.2 Objetivos específicos

Necesitamos elegir el dataset con el que será entrenado el algoritmo. Asimismo, requerimos de limpiar los datos de nuestro dataset para poder hacer un mejor procesado de la información. Y, por último, utilizaremos algún titular de un periódico confiable para hacer nuestra primera predicción.

0.5 3. Justificación

Nos proponemos entonces investigar cómo funciona el algoritmo de Naive Bayes y utilizaremos los conocimientos adquiridos para implementarlo a nuestro problema previamente mencionado, así mismo, trabajaremos sobre la plataforma de Anaconda3, específicamente en la herramienta denominada Jupyter Notebooks.

0.6 4. Aplicación de la metodología

0.6.1 4.1 Comprensión del negocio

Lo que realmente se quiere emplear este trabajo con la ayuda de este clasificador, como ya lo hemos mencionado, es que los usuarios o clientes no obtengan información falsa y al igual ellos mismos no puedan compartir esa información, así que con la herramienta ocupada clasificaremos lo que probablemente sea fake y se haga un stop de esa información.

0.7 Lo primero que haremos sera cargar las librerias indicadas para poder manejar los datos necesarios

```
[1]: import pandas as pd
import numpy as np
import string
import time
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
from nltk.corpus import stopwords
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
import seaborn as sns
```

0.7.1 4.2 Comprensión de los datos

El dataset usado en este proyecto provienen de los dataset “Fake and real news dataset” extraídos de la página Kaggle.com. El primer dataset nombrado “Fake” contiene 23,503 registros de noticias clasificadas como falsas, que están subdivididos en 6 categorías las cuales son: News, Politics,

Government News, Loft News, US News y Middle East. De manera similar, el segunda dataset nombrado “True” contiene 21,418 registros que están divididos en 2 categorías: Politic News y World News. Cada dataset contiene 5 columnas: title=título de la noticia, text=cuerpo de la noticia, subject=clasificación de la noticia y date=fecha de publicación de la noticia.

0.7.2 Los datasets son obtenidos de Kaggle (Fake and real news dataset)

```
[2]: fake = pd.read_csv("Fake.csv")
      true = pd.read_csv("True.csv")
```

0.7.3 Imprimiremos los datos de el dataset llamado Fake

```
[3]: fake.head()
```

```
[3]:
```

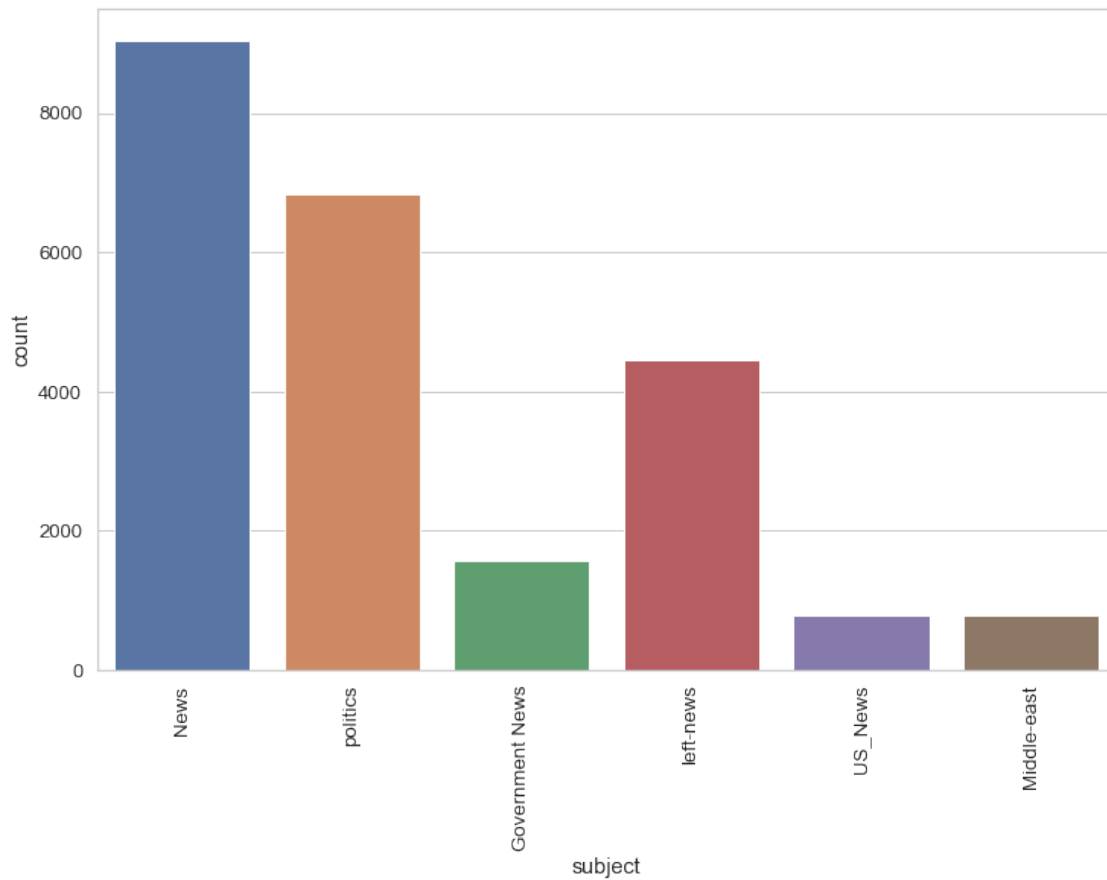
	title \	text	subject \	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017

0.7.4 Para darnos una idea de la clasificacion de noticias sobre las que mas se hacen noticias falsas, haremos una grafica.

```
[4]: plt.figure(figsize = (12,8))
      sns.set(style = "whitegrid",font_scale = 1.2)
      chart = sns.countplot(x = "subject" , data = fake)
      chart.set_xticklabels(chart.get_xticklabels(),rotation=90)
```

```
[4]: [Text(0, 0, 'News'),
      Text(0, 0, 'politics'),
      Text(0, 0, 'Government News'),
      Text(0, 0, 'left-news'),
```

```
Text(0, 0, 'US_News'),
Text(0, 0, 'Middle-east')]
```



0.7.5 Como se puede ver en la grafica, las noticias sobre las que mas se hacen noticias falsas es sobre noticias en general, seguida de noticias sobre politica.

0.7.6 Imprimiremos los primeros datos de el dataset llamado True

```
[5]: true.head()
```

```
[5]:                                     title \
0  As U.S. budget fight looms, Republicans flip t...
1  U.S. military to accept transgender recruits o...
2  Senior U.S. Republican senator: 'Let Mr. Muell...
3  FBI Russia probe helped by Australian diplomat...
4  Trump wants Postal Service to charge 'much mor...

                                     text      subject \
0  WASHINGTON (Reuters) - The head of a conservat...  politicsNews
```

```

1 WASHINGTON (Reuters) - Transgender people will... politicsNews
2 WASHINGTON (Reuters) - The special counsel inv... politicsNews
3 WASHINGTON (Reuters) - Trump campaign adviser ... politicsNews
4 SEATTLE/WASHINGTON (Reuters) - President Donal... politicsNews

```

```

      date
0  December 31, 2017
1  December 29, 2017
2  December 31, 2017
3  December 30, 2017
4  December 29, 2017

```

0.7.7 Para darnos una idea de la clasificacion de noticias sobre las que mas se hacen noticias veridicas, haremos una grafica.

```

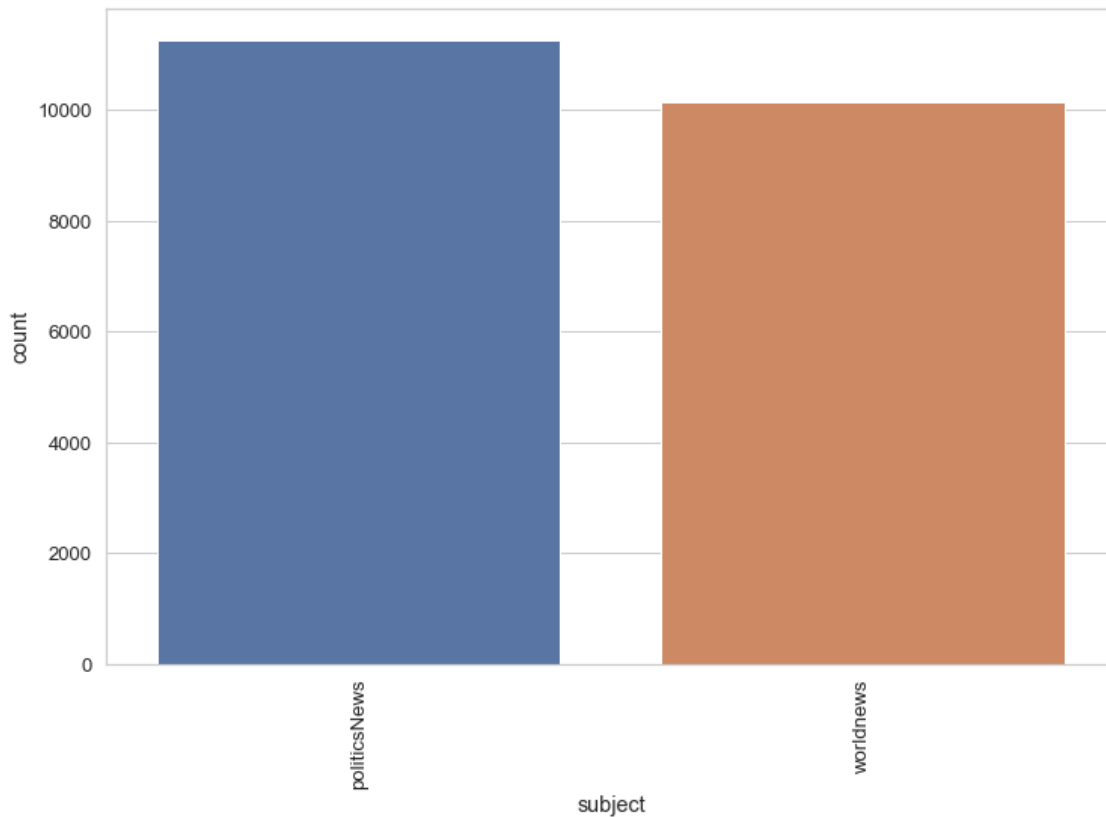
[6]: plt.figure(figsize = (12,8))
     sns.set(style = "whitegrid",font_scale = 1.2)
     chart = sns.countplot(x = "subject" , data = true)
     chart.set_xticklabels(chart.get_xticklabels(),rotation=90)

```

```

[6]: [Text(0, 0, 'politicsNews'), Text(0, 0, 'worldnews')]

```



0.7.8 Como podemos ver, solo existen dos tipos de clasificaciones de noticias verdaderas.

0.7.9 Incluiremos en ambos dataset, una columna llamada category, en el dataset fake a la columna le asignaremos el valor 0 y correspondientemente a el dataset true le asignaremos el valor 1.

```
[7]: true['category']=1
     fake['category']=0
```

0.7.10 Crearemos un nuevo dataset donde incluiremos los dos dataset anteriores para poder manejar mejor los datos.

```
[8]: df = pd.concat([true,fake])
```

0.7.11 Eliminaremos la columna date, ya que en este caso no la necesitaremos para entrenar nuestro algoritmo, asi mismo, eliminaremos del dataset las filas donde se encuentre por lo menos un valor nulo de alguna columna.

```
[9]: df= df.drop(['date'], axis=1)
     df = df.dropna()
```

0.7.12 Para comprobar que el dataframe se ah creado de manera correcta, lo mostraremos en pantalla.

```
[10]: df
```

```
[10]:
```

		title \		
0	As U.S. budget fight looms, Republicans flip t...			
1	U.S. military to accept transgender recruits o...			
2	Senior U.S. Republican senator: 'Let Mr. Muell...			
3	FBI Russia probe helped by Australian diplomat...			
4	Trump wants Postal Service to charge 'much mor...			
...				
23476	McPain: John McCain Furious That Iran Treated ...			
23477	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...			
23478	Sunnistan: US and Allied 'Safe Zone' Plan to T...			
23479	How to Blow \$700 Million: Al Jazeera America F...			
23480	10 U.S. Navy Sailors Held by Iranian Military ...			
		text	subject \	
0	WASHINGTON (Reuters) - The head of a conservat...		politicsNews	
1	WASHINGTON (Reuters) - Transgender people will...		politicsNews	
2	WASHINGTON (Reuters) - The special counsel inv...		politicsNews	
3	WASHINGTON (Reuters) - Trump campaign adviser ...		politicsNews	
4	SEATTLE/WASHINGTON (Reuters) - President Donal...		politicsNews	
...		

23476	21st Century Wire says As 21WIRE reported earl...	Middle-east
23477	21st Century Wire says It s a familiar theme. ...	Middle-east
23478	Patrick Henningsen 21st Century WireRemember ...	Middle-east
23479	21st Century Wire says Al Jazeera America will...	Middle-east
23480	21st Century Wire says As 21WIRE predicted in ...	Middle-east

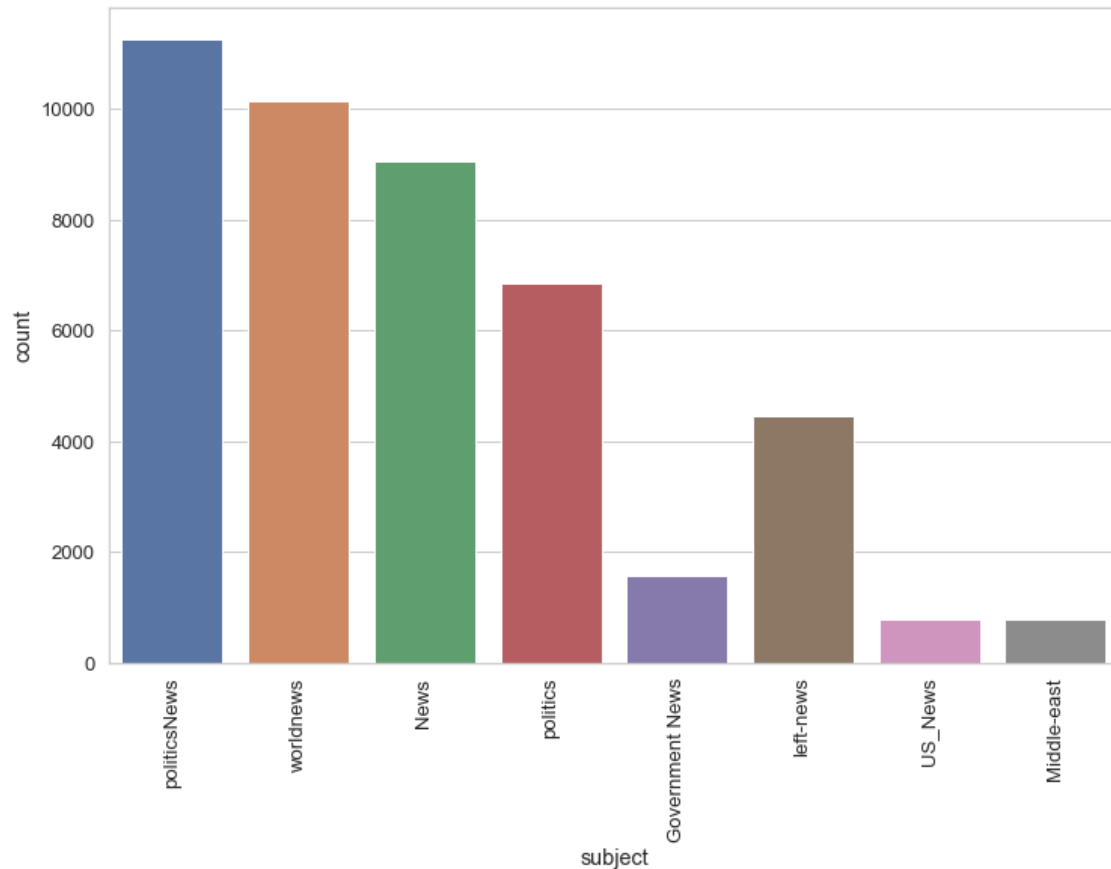
	category
0	1
1	1
2	1
3	1
4	1
...	...
23476	0
23477	0
23478	0
23479	0
23480	0

[44898 rows x 4 columns]

0.7.13 Mostraremos nuevamente la grafica para poder ver de una mejor manera los datos y poder interpretarlos.

```
[11]: plt.figure(figsize = (12,8))
sns.set(style = "whitegrid",font_scale = 1.2)
chart = sns.countplot(x = "subject", data = df)
chart.set_xticklabels(chart.get_xticklabels(),rotation=90)
```

```
[11]: [Text(0, 0, 'politicsNews'),
Text(0, 0, 'worldnews'),
Text(0, 0, 'News'),
Text(0, 0, 'politics'),
Text(0, 0, 'Government News'),
Text(0, 0, 'left-news'),
Text(0, 0, 'US_News'),
Text(0, 0, 'Middle-east')]
```



0.7.14 4.3 Preparación de los datos

Para obtener una mejor clasificación de los datos, necesitamos primero, limpiarlos y es por eso que necesitamos limpiar los datos. Lo primero que se hace es remover de las filas que contienen datos vacíos, una vez que tenemos los datos que tienen todas las filas llenas, requerimos limpiar los signos de puntuación de las oraciones, ahora necesitamos eliminar las palabras que no tienen un significado por si solas, denominadas StopWords.

0.7.15 Esta celda la necesitamos para poder limpiar los datos:

- Con la primera instruccion, separaremos las cadenas en caracteres y eliminaremos los signos de puntuación.
- En la segunda instruccion uniremos los caracteres.
- Por ultimo, haremos que todas las palabras sean minusculas y al mismo tiempo eliminaremos las palabras que no tienen un significado por si solas, denominadas StopWords.

```
[12]: def text_cleaning(text):
    remove_punctuation = [char for char in text if char not in string.
        ↳ punctuation]
    remove_punctuation = ''.join(remove_punctuation)
    return [word for word in remove_punctuation.split() if word.lower() not in
        ↳ stopwords.words('english')]
```


0.8 En esta celda, se asigna a una variable el texto de la columna title

```
[13]: t1 = time.time()
title_cleaning = df.iloc[:,0].apply(text_cleaning)
t2 = time.time()
print("Time to clean ", len(df), "articles:", (t2-t1)/60, "min")
text_cleaning
```

Time to clean 44898 articles: 4.738045958677928 min

```
[13]: <function __main__.text_cleaning(text)>
```

0.9 Como la variable title_cleaning es una lista que incluye todos los titulos de cada renglon separados en palabras, necesitaremos unir estas palabras en oraciones para poder aplicar el algoritmo.

```
[14]: def listToString(s):
      str1 = " "
      return (str1.join(s))
```

```
[15]: title_cleaning = title_cleaning.apply(listToString)
```

```
[16]: title_cleaning
```

```
[16]: 0      US budget fight looms Republicans flip fiscal ...
1      US military accept transgender recruits Monday...
2      Senior US Republican senator Let Mr Mueller job
3      FBI Russia probe helped Australian diplomat ti...
4      Trump wants Postal Service charge much Amazon ...
...
23476   McPain John McCain Furious Iran Treated US Sai...
23477   JUSTICE Yahoo Settles Email Privacy Classactio...
23478   Sunnistan US Allied 'Safe Zone' Plan Take Terr...
23479   Blow 700 Million Al Jazeera America Finally Ca...
23480   10 US Navy Sailors Held Iranian Military - Sig...
Name: title, Length: 44898, dtype: object
```

0.9.1 4.4 Modelado

Una de las aplicaciones principales de Machine Learning, es la detección de spam. Casi todos los servicios de email más importantes proporcionan un detector de spam que clasifica el spam automáticamente y lo envía al buzón de correo no deseado.

0.9.2 Esta instruccion sirve para poder dividir nuestro dataframe en dos partes:

x_train es la variable donde se almacenaran los titulares de las noticias que entrenara al alg
x_test esta variable de la misma manera, almacenara los titulares de las noticias pero esta no
y_train es la variable donde se almacenaran los valores objetivos con las que se entrenara el a

y_test esta variable de la misma manera, almacenaran los valores objetivos pero estos nos servirán. title_cleaning representa la variable que se separara para poder asignarla a cada variable x y y_test indica el tamaño en el que se dividira el dataset para poder testear los algoritmos. En este caso, utilizaremos el 20% del dataset para testear y el 80% para entrenar el algoritmo

```
[17]: x_train, x_test, y_train, y_test = train_test_split(title_cleaning, df.  
    ↪category, test_size=0.2)
```

0.9.3 vect es una instancia de la clase CountVectorizer. vect.fit_transform convierte las características de entrenamiento en vectores numericos. vect.transform convierte las etiquetas de entrenamiento en vectores numericos.

```
[18]: vect = CountVectorizer(max_features=5000)  
X_train = vect.fit_transform(x_train)  
X_test = vect.transform(x_test)
```

0.9.4 4.5 Evaluación

Se evaluará implementando las herramientas de Python y el uso de probabilidad. Convirtiendo los datos en una tabla de frecuencias para después crear una tabla de probabilidad calculando las correspondientes a que ocurran los diversos eventos. La ecuación de Naive Bayes se usa para calcular la probabilidad posterior de cada clase y con la clase posterior mas alta es el resultado de la predicción.

0.9.5 En esta parte creamos una instancia de la clase MultinomialNB (multinomial naive bayes) que sera la encargada de procesar los datos.

- mnb es la instancia de la clase
- mnb.fit entrenara el algoritmo, recibe los parametros con los que sera entrenado, X_train es
- mnb.predict aplica el algoritmo entrenado previamente para predecir los resultados de cada e

```
[19]: mnb = MultinomialNB(alpha =0.2)  
  
mnb.fit(X_train,y_train)  
  
result= mnb.predict(X_test)  
print(result)
```

```
[0 1 0 ... 1 0 1]
```

0.9.6 Esta instruccion nos da la exactitud de nuestro modelo.

```
[20]: accuracy_score(result,y_test)
```

```
[20]: 0.9385300668151447
```

0.9.7 Esta funcion sirve para poder analizar diferentes entradas y poder predecir si la entrada representa el titular de una noticia fake o real.

```
[21]: def predict_news(news):  
    test = vect.transform(news)  
    pred= mnbs.predict(test)  
    if pred == 1:  
        return 'True'  
    elif pred == 0:  
        return 'Fake'  
    else:  
        return 'no class found'
```

0.9.8 X recibe el titular de alguna noticia y se utiliza la funcion predict_news para poder predecir si la noticia es falsa o verdadera. En el ejemplo se muestra que dado el titular de una noticia real obtenida del sitio web The New York Times y es capaz de clasificarla como real.

```
[23]: x=["California Tax Revolt Faces a Retreat, 40 Years Later"]  
r = predict_news(x)  
print (r)
```

True

0.10 5. Conclusión

Con el clasificador, tendremos una facilidad de implementar la información que es fake y es una herramienta que funciona muy bien, ajustar sus parámetros será raramente necesario y nos tenemos que centrar en la precisión y rapidez del procesamiento y ejecución de los resultados.

0.11 6. Bibliografía

Mitchell, Tom, Machine Learning, Ed. McGraw-Hill (1997). <https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fundamentos-e-implementaci%C3%B3n-4bcb24b307f>
<https://rpubs.com>